

[illegible][illegible]

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

[illegible]

H 10
16-Sep-1984 00:16:37
5-Sep-1984 14:24:01

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]TU81SENSE.FOR;1

Page 1

```
0001      Subroutine TU81_SENSE_BYTES_DECODE (lun)
0002      C
0003      C Version:      'V04-000'
0004      C
0005      C*****
0006      C*
0007      C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008      C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009      C* ALL RIGHTS RESERVED.
0010      C*
0011      C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012      C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013      C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014      C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015      C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016      C* TRANSFERRED.
0017      C*
0018      C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019      C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020      C* CORPORATION.
0021      C*
0022      C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023      C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024      C*
0025      C*
0026      C*****
0027      C
0028      C      Author: Sharon Reynolds      Creation Date: 5-Jul-1984
0029      C
0030      C      Description:
0031      C
0032      C          This routine decodes the device dependent information
0033      C          for the TU81 that is returned with a 'tape tranfer error'
0034      C          packet.
0035      C
0036      C--
0037      C**
0038
0039      Include 'SRC$:MSGHDR /nolist'
0098
0099
0100      Byte      lun, code, extended_sense(0:15)
0101
0102      Logical*1      bit
0103
0104      Integer*4      COMPRESS4
0105
0106      Equivalence      (emb(82),extended_sense)
0107
0108      Character*22      byte_0(0:7)
0109      Data      byte_0(0)      /*UNIT EXCEPTION*/
0110      Data      byte_0(1)      /*UNIT CHECK*/
0111      Data      byte_0(3)      /*DATA CHECK*/
0112      Data      byte_0(4)      /*EQUIPMENT CHECK*/
0113      Data      byte_0(6)      /*INTERVENTION REQUIRED*/
0114      Data      byte_0(7)      /*COMMAND REJECT*/
0115
```

```
0116 Character*24 byte_1(0:7)
0117 Data byte_1(0) /*DEVICE OFFLINE*/
0118 Data byte_1(1) /*DEVICE NOT READY*/
0119 Data byte_1(3) /*RESET KEY*/
0120 Data byte_1(4) /*FILE PROTECTED*/
0121 Data byte_1(6) /*DEVICE COMMAND CHECK*/
0122 Data byte_1(7) /*ILLEGAL CHANNEL COMMAND*/
0123
0124 Character*23 byte_2(0:7)
0125 Data byte_2(0) /*DEVICE INTERRUPT CHECK*/
0126 Data byte_2(1) /*VELOCITY CHECK*/
0127 Data byte_2(2) /*DEVICE HARDWARE CHECK*/
0128 Data byte_2(3) /*DEVICE RESPONSE CHECK*/
0129 Data byte_2(4) /*WRITE HARDWARE CHECK*/
0130 Data byte_2(5) /*READ HARDWARE CHECK*/
0131 Data byte_2(6) /*CHANNEL RESPONSE CHECK*/
0132 Data byte_2(7) /*CHANNEL PARITY ERROR*/
0133
0134 Character*14 byte_3_prt1(0:2)
0135 Data byte_3_prt1(0) /*TAPE MOVED*/
0136 Data byte_3_prt1(1) /*BOT*/
0137 Data byte_3_prt1(2) /*UNRECOVERABLE*/
0138
0139 Character*16 byte_3_prt2(4:7)
0140 Data byte_3_prt2(4) /*ID FAULT*/
0141 Data byte_3_prt2(5) /*READ DATA CHECK*/
0142 Data byte_3_prt2(6) /*PE CRC CHECK*/
0143 Data byte_3_prt2(7) /*AGC FAULT*/
0144
0145 Character*26 byte_5(0:7)
0146 Data byte_5(0) /*VARIABLE GAP MODE (SHORT)*/
0147 Data byte_5(1) /*VARIABLE GAP MODE (LONG)*/
0148 Data byte_5(2) /*START/STOP MODE*/
0149 Data byte_5(3) /*DIAGNOSTIC MODE*/
0150 Data byte_5(4) /*FILE MARK DETECTED*/
0151 Data byte_5(5) /*AUTO SPEED MODE*/
0152 Data byte_5(6) /*HIGH SPEED MODE*/
0153 Data byte_5(7) /*GCR MODE*/
0154
0155 Character*33 byte_6(2:7)
0156 Data byte_6(2) /*WRITE AUXILIARY CRC PARITY ERROR*/
0157 Data byte_6(4) /*45 PARITY ERROR*/
0158 Data byte_6(5) /*WRITE CRC PARITY ERROR*/
0159 Data byte_6(6) /*RESIDUAL BYTE COUNT CHECK*/
0160 Data byte_6(7) /*WRITE TRANSFER CHECK*/
0161
0162 Character*25 byte_7(0:7)
0163 Data byte_7(0) /*EXCESSIVE POINTERS*/
0164 Data byte_7(1) /*NO TRACK POINTER*/
0165 Data byte_7(2) /*UNCORRECTABLE DATA*/
0166 Data byte_7(3) /*ECC3 CHECK*/
0167 Data byte_7(4) /*RESYNC CHECK*/
0168 Data byte_7(5) /*READ AUXILIARY CRC CHECK*/
0169 Data byte_7(6) /*READ CRC CHECK*/
0170 Data byte_7(7) /*RESIDUAL CHECK*/
0171
0172 Character*22 byte_8(0:7)
```



```
0173      Data      byte_8(0)      /*'ARA ID CHECK*/  
0174      Data      byte_8(1)      /*'ARA BURST CHECK*/  
0175      Data      byte_8(2)      /*'ID CHECK*/  
0176      Data      byte_8(3)      /*'WRITE TAPE MARK CHECK*/  
0177      Data      byte_8(4)      /*'READ TIME OUT*/  
0178      Data      byte_8(5)      /*'SKEW ERROR*/  
0179      Data      byte_8(6)      /*'POSTAMBLE ERROR*/  
0180      Data      byte_8(7)      /*'NOISE CHECK*/  
0181  
0182      Character*29      byte_9(0:7)  
0183      Data      byte_9(0)      /*'TRACK IN ERROR PARITY BIT = */  
0184      Data      byte_9(1)      /*'SINGLE TRACK CORRECTABLE*/  
0185      Data      byte_9(2)      /*'DUAL TRACK CORRECTABLE*/  
0186      Data      byte_9(3)      /*'END MARK CHECK*/  
0187      Data      byte_9(4)      /*'READ DATA PARITY ERROR*/  
0188      Data      byte_9(5)      /*'READ TRANSFER CHECK*/  
0189      Data      byte_9(6)      /*'READ BUFFER IN PARITY ERROR*/  
0190      Data      byte_9(7)      /*'EC HARDWARE CHECK*/  
0191  
0192      Character*1      byte_10(0:7)  
0193      Data      byte_10(0)      /*'2*/  
0194      Data      byte_10(1)      /*'8*/  
0195      Data      byte_10(2)      /*'1*/  
0196      Data      byte_10(3)      /*'9*/  
0197      Data      byte_10(4)      /*'3*/  
0198      Data      byte_10(5)      /*'5*/  
0199      Data      byte_10(6)      /*'6*/  
0200      Data      byte_10(7)      /*'7*/  
0201  
0202      Character*15      byte_11(0:7)  
0203      Data      byte_11(0)      /*'EOT*/  
0204      Data      byte_11(1)      /*'BOT*/  
0205      Data      byte_11(2)      /*'HIGH SPEED*/  
0206      Data      byte_11(3)      /*'GAP CONTROL*/  
0207      Data      byte_11(4)      /*'FILE PROTECTED*/  
0208      Data      byte_11(5)      /*'REWIND*/  
0209      Data      byte_11(6)      /*'ONLINE*/  
0210      Data      byte_11(7)      /*'READY*/  
0211  
0212      Character*9      byte_12(2:7)  
0213      Data      byte_12(2)      /*'S/S MODE*/  
0214      Data      byte_12(3)      /*'LONG GAP*/  
0215      Data      byte_12(4)      /*'GCR*/  
0216      Data      byte_12(5)      /*'DSE*/  
0217      Data      byte_12(6)      /*'WRITE*/  
0218      Data      byte_12(7)      /*'REVERSE*/  
0219  
0220      Character*22      byte_13(0:7)  
0221      Data      byte_13(0)      /*'AIR FLOW/TEMP CHECK*/  
0222      Data      byte_13(1)      /*'REVERSE IN BOT*/  
0223      Data      byte_13(2)      /*'RESET KEY*/  
0224      Data      byte_13(3)      /*'AGC CHECK*/  
0225      Data      byte_13(4)      /*'DENSITY CHANGE*/  
0226      Data      byte_13(5)      /*'DIAGNOSTIC REQUEST*/  
0227      Data      byte_13(6)      /*'INTERVENTION REQUIRED*/  
0228      Data      byte_13(7)      /*'COMMAND REJECT*/  
0229
```

```

0230 C Output the extended sense information header.
0231 C
0232 Write (lun,5) 'TU81 EXTENDED SENSE INFORMATION'
0233 5 Format (' ',A31,/)
0234
0235
0236 C Decode and output byte 0 of the extended sense information.
0237 C
0238 Write (lun,10) 'BYTE 0', extended_sense(0)
0239 10 Format (' ',T8,A,T24,Z2.2)
0240
0241 Call OUTPUT (lun,extended_sense(0),byte_0, 0, 0, 7, '0')
0242
0243
0244 C Decode and output byte 1 of the extended sense information.
0245 C
0246 Write (lun,10) 'BYTE 1', extended_sense(1)
0247 Call OUTPUT (lun,extended_sense(1),byte_1, 0, 0, 7, '0')
0248
0249 C Decode and output byte 2 of the extended sense information.
0250 C
0251 Write (lun,10) 'BYTE 2', extended_sense(2)
0252 Call OUTPUT (lun,extended_sense(2),byte_2, 0, 0, 7, '0')
0253
0254
0255 C Decode and output byte 3 of the extended sense information.
0256 C
0257 Write (lun,10) 'BYTE 3', extended_sense(3)
0258
0259 Call OUTPUT (lun,extended_sense(3),byte_3_prt1, 0, 0, 2, '0')
0260 Call OUTPUT (lun,extended_sense(3),byte_3_prt2, 4, 4, 7, '0')
0261
0262
0263 C Decode and output byte 4 of the extended sense information.
0264 C
0265 Write (lun,10) 'BYTE 4', extended_sense(4)
0266
0267 Write (lun,20) 'FORMATTER COMMAND CODE = ', extended_sense(4)
0268 20 Format (' ',T40,A25,Z2.2,'(X)')
0269
0270
0271 C Decode and output byte 5 of the extended sense information.
0272 C
0273 Write (lun,10) 'BYTE 5', extended_sense(5)
0274 Call OUTPUT (lun,extended_sense(5),byte_5, 0, 0, 7, '0')
0275
0276
0277 C Decode and output byte 6 of the extended sense information.
0278 C
0279 Write (lun,10) 'BYTE 6', extended_sense(6)
0280 Call OUTPUT (lun,extended_sense(6),byte_6, 2, 2, 7, '0')
0281
0282
0283 C Decode and output byte 7 of the extended sense information.
0284 C
0285 Write (lun,10) 'BYTE 7', extended_sense(7)
0286 Call OUTPUT (lun,extended_sense(7),byte_7, 0, 0, 7, '0')

```



```
0287
0288
0289 C Decode and output byte 8 of the extended sense information.
0290 C
0291     Write (lun,10) 'BYTE 8', extended_sense(8)
0292     Call OUTPUT (lun,extended_sense(8),byte_8, 0, 0, 7, '0')
0293
0294
0295 C Decode and output byte 9 of the extended sense information.
0296 C
0297     Write (lun,10) 'BYTE 9', extended_sense(9)
0298     Call OUTPUT (lun,extended_sense(9),byte_9, 0, 0, 7, '0')
0299
0300
0301 C Decode and output byte 10 of the extended sense information.
0302 C
0303     Write (lun,25) 'BYTE 10', extended_sense(10)
0304 25     Format (' ',T8,A,T24,Z2.2)
0305
0306     If (extended_sense(10) .NE. 0) then
0307
0308         Do 27, I=0,7
0309
0310             Bit = LIB$EXTZV (I,1,extended_sense(10))
0311             If (bit) then
0312                 Call LINCHK (lun,1)
0313                 Write (lun,26) 'TRACK IN ERROR = ', byte_10(I)
0314 26         Format (' ',T40,A17,A1,'.')
0315             Endif
0316
0317 27         Continue
0318         Endif
0319
0320 C Decode and output byte 11 of the extended sense information.
0321 C
0322     Write (lun,10) 'BYTE 11', extended_sense(11)
0323     Call OUTPUT (lun,extended_sense(11),byte_11, 0, 0, 7, '0')
0324
0325
0326 C Decode and output byte 12 of the extended sense information.
0327 C
0328     Write (lun,10) 'BYTE 12', extended_sense(12)
0329     Call OUTPUT (lun,extended_sense(12),byte_12, 2, 2, 7, '0')
0330
0331
0332 C Decode and output byte 13 of the extended sense information.
0333 C
0334     Write (lun,10) 'BYTE 13', extended_sense(13)
0335     Call OUTPUT (lun,extended_sense(13),byte_13, 0, 0, 7, '0')
0336
0337
0338 C Decode and output byte 14 of the extended sense information.
0339 C
0340     Write (lun,28) 'BYTE 14', extended_sense(14),'COMMAND CODE'
0341 28     Format (' ',T8,A,T24,Z2.2,/,T40,A12)
0342
0343 C Decode and output byte 15 of the extended sense information.
```

```
0344      C
0345      Write (lun,30) 'BYTE 15', extended_sense(15), 'MARGINAL CONDITION CODE'
0346      30      Format (' ',T8,A,T24,Z2.2,/,T40,A23)
0347
0348
0349      End
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	1142	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	327	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	3084	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	5065	

ENTRY POINTS

Address	Type	Name
0-00000000		TU81_SENSE_BYTES_DECODE

VARIABLES

Address	Type	Name	Address	Type	Name
2-000007EF	L*1	BIT	2-000007EE	L*1	CODE
2-000007F0	I*4	COMPRESS4	3-00000000	I*4	EMB\$SL_HD_SID
3-00000004	I*2	EMB\$W_HD_ENTRY	3-0000000E	I*2	EMB\$W_HD_ERRSEQ
2-000007F4	I*4	I	AP-00000004	L*1	LUN

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	CHAR	BYTE_0	176	(0:7)
2-00000080	CHAR	BYTE_1	192	(0:7)
2-00000688	CHAR	BYTE_10	8	(0:7)
2-00000690	CHAR	BYTE_11	120	(0:7)
2-00000708	CHAR	BYTE_12	54	(2:7)
2-0000073E	CHAR	BYTE_13	176	(0:7)
2-00000170	CHAR	BYTE_2	184	(0:7)
2-00000228	CHAR	BYTE_3_PRT1	42	(0:2)
2-00000252	CHAR	BYTE_3_PRT2	64	(4:7)
2-00000292	CHAR	BYTE_5	208	(0:7)
2-00000362	CHAR	BYTE_6	198	(2:7)
2-00000428	CHAR	BYTE_7	200	(0:7)
2-000004F0	CHAR	BYTE_8	176	(0:7)
2-000005A0	CHAR	BYTE_9	232	(0:7)

TU81_SENSE_BYTES_DECODE

N 10
16-Sep-1984 00:16:37
5-Sep-1984 14:24:01VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]TU81SENSE.FOR;1
Page 73-00000000 L*1 EMB
3-00000006 I*4 EMB\$Q HD_TIME
3-00000052 L*1 EXTENDED_SENSE512 (0:511)
8 (2)
16 (0:15)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-000000E8	5'	1-000000F0	10'	1-000000FC	20'	1-0000010C	25'	1-00000118	26'	**	27
1-00000125	28'	1-00000136	30'								

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	LIB\$EXTZV		LINCHK		OUTPUT

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:TU81SENSE/OBJ=OBJ\$:TU81SENSE MSRC\$:TU81SENSE

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE_FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 4.30 seconds
Elapsed Time: 13.08 seconds
Page Faults: 169
Dynamic Memory: 185 pages

0154

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY